UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/039,289	01/04/2002	Gilbert Wolrich	042390.P12851	8345

45209 7590 06/29/2010

INTEL/BSTZ
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
CHANNAVAJJALA, SRIRAMA T	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
06/29/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte GILBERT WOLRICH, MARK B. ROSENBUTH,
and DEBRA BERNSTEIN

Appeal 2009-004531
Application 10/039,289
Technology Center 2100

Decided: June 29, 2010

Before JOHN A. JEFFERY, ST. JOHN COURTENAY, III, and STEPHEN C. SIU, *Administrative Patent Judges*.

COURTENAY, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellants seek our review under 35 U.S.C. § 134 of the Examiner's decision rejecting claims 1-37. We have jurisdiction over the appeal under 35 U.S.C. § 6(b). We Affirm-in-part.

Invention

Appellants invention on appeal is directed to utilizing queue arrays in network devices to handle fast network line speeds. (Spec. 1).

Claim 1 is illustrative:

1. A method comprising:

storing in memory a plurality of queue descriptors each including a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue;

in response to a command to perform an enqueue or dequeue operation with respect to a first queue, fetching from the memory to a cache one of either the head pointer or tail pointer of a first queue descriptor corresponding to the first queue; and

returning to the memory from the cache portions of the first queue descriptor modified by the operation.

Appellants appeal the following rejection:

1. Claims 1-37 under 35 U.S.C. § 103(a) as unpatentable over Pascal (EP 0760501 A1, Mar. 5, 1997) and Slane (US 6,438,651 B1, Aug. 20, 2002).

ISSUES

Based upon our review of the administrative record, we have determined that the following issues are dispositive in this appeal:

1. Did the Examiner err in finding under § 103 that the combination of Pascal and Slane would have taught or suggested a plurality of queue descriptors and fetching

from a memory to a cache either a head or tail pointer of a first queue descriptor within the meaning of representative claim 1?

2. Did the Examiner err in finding under § 103 that the combination of Pascal and Slane would have taught or suggested fetching the head pointer and not the tail pointer of the first queue descriptor in response to a command to perform a dequeue operation with respect to the first queue? (Representative claim 2).

3. Did the Examiner err in finding under § 103 that the combination of Pascal and Slane would have taught or suggested returning to memory the head pointer and not the tail pointer of the first queue descriptor if only dequeue operations were performed on the first queue? (Representative claim 4).

4. Did the Examiner err by improperly combining the Pascal and Slane references under § 103?

5. Did the Examiner err in finding under § 103 that the combination of Pascal and Slane would have taught or suggested fetching from the memory to a cache the count and one of either the head pointer or tail pointer within the meaning of representative claim 31?

FACTUAL FINDINGS

1. Pascal teaches the use of a head pointer 32 and tail pointer 30 where the head pointer points to the next queue location to be read and the tail pointer points to the next queue location to written. (Col. 1, ll. 21-33; col. 7, ll. 18-20; Fig. 3).
2. Pascal teaches that consumer index (C-index) 37 points to the next queue location to be read. (Col. 8, ll. 13-15; Fig. 3).
3. Pascal teaches that producer index (P-index) 34 points to the next queue location to be written. (Col. 8, ll. 4-6); Fig. 3).
4. Pascal teaches that "queue 10 is said to be empty when the P-index and C-index represent the same position in the continuum 40 (this, in turn, corresponding to the head and tail pointers pointing to the same storage location in queue 10)." (Col. 8, ll. 34-38).
5. Pascal teaches that a copy of the consumer index (C-index) 37 is stored in register 36. (Col. 8, ll. 22-23; Fig. 3).
6. Pascal teaches that a copy of the producer index (P-index) 34 is stored in register 39. (Col. 8, ll. 23-25; Fig. 3).
7. Pascal teaches the use of two count registers 31 and 33, where count register 31 indicates the number of available queue storage locations to be written to and count register 33 indicates the number of available queue storage locations to be accessed (read). (Col. 7, ll. 33-39 and 51-54; Fig. 3).
8. Slane teaches:

> As shown in FIG. 2, the queue user 52 performs enqueue and dequeue operations to the queue 62. The enqueue operation places new data at the tail 66 of the queue 62 and therefore does a memory write. The dequeue

> operation removes data from the head 64 of the queue 62, and therefore does a memory read. These memory reads and writes are presented to the cache 54 to fulfill. The cache 54 remains transparent to the queue user 52 that performs enqueue/dequeue operations to the circular buffer 52, which are translated into read/write operations to the cache 54. In this way, an entity that wants to access, i.e., read/dequeue or write/enqueue, data to the circular buffer 62, would request the data and the queue user 52 would return the data from the cache 54 much faster than the process of accessing and returning the data from the main memory 62 over the bus 68.

(Col. 3, ll. 52-66).

9. Slane expressly teaches the use of head and tail queue pointers. (Col. 4, l. 45; Fig. 2).

Grouping of Claims

Appellants argue claims 1, 7-14, 20-26, and 29-30 as a group (App. Br. 9 *et seq.*). We select representative claim 1 to decide the appeal for this group.

Appellants argue claims 2, 3, 15, 16, and 27 as a group (App. Br. 13). We select representative claim 2 to decide the appeal for this group.

Appellants argue claims 4-6, 17-19, and 28 as a group. (App. Br. 13 *et seq.*). We select representative claim 4 to decide the appeal for this group. *See* 37 C.F.R. § 41.37(c)(1)(vii).

We consider claims 31-37 separately *infra*.

ANALYSIS

Issue 1

We decide the question of whether the Examiner erred by finding that the combination of Pascal and Slane would have taught or suggested a plurality of queue descriptors and fetching from a memory to a cache either a head or tail pointer of a first queue descriptor within the meaning of representative claim 1.

At the outset, we find Pascal clearly teaches at least a first "queue descriptor" including a head pointer and a tail pointer within the meaning of claim 1. (*See* FF 1, Fig. 1). We observe that Slane also expressly teaches the use of a set of head and tail queue pointers. (FF 9).

However, as argued by Appellants, the claim requires a plurality of queue descriptors, with each queue descriptor including a head pointer and a tail pointer. (App. Br. 9).

Upon closely inspecting Pascal's Figure 3 and the supporting description sections (FF 2-3), we observe that Pascal teaches a consumer index 37 (C-index) that points to the next queue location to be read. (FF 2). Thus, we find that C-index 37 contains the same read pointer (address) value as head pointer 32. Likewise, we find P-index 34 points to the next queue location to be written. (FF 2). Therefore, we find Pascal's P-index 34 and C-index 37 correspond to tail and head pointers 30 and 32, respectively. (*cf.* FF 4).

While Pascal's P and C-indexes are used for determining queue status (e.g., queue empty or queue full conditions),[1] we also find these index registers (P-index 34 and C-index 37) teach or suggest the claim requirement for a second (plurality) of "*queue descriptors* each including a *head pointer* [C-index 37] pointing to a first element in a corresponding queue and a *tail pointer* [P-index 34] pointing to a last element in the corresponding queue." (Claim 1, emphasis added). We also find the addition of a second "queue descriptor" in Appellants' claim 1 is nothing more than a "predictable use of prior art elements according to their established functions" and, as such, we find claiming a mere plurality of prior art elements (*queue descriptors* each including a *head pointer* and a *tail pointer*) is not a patentable distinction over the prior art of record. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, (2007).

We also find that the Examiner's cited combination of Pascal and Slane would have taught or suggested "in response to a command to perform an enqueue or dequeue operation with respect to a first queue, *fetching from the memory to a cache one of either the head pointer or tail pointer* of a first queue descriptor corresponding to the first queue." (Claim 1, emphasis added). In particular, we find Pascal teaches that a ("fetched") copy of the consumer index (C-index) 37 (i.e., a head pointer) is stored in register 36. (FF 5). We find that Pascal also teaches that a (fetched) copy of the producer index (P-index) 34 (i.e., a tail pointer) is stored in register 39. (FF 6). We find Pascal's index copies (36 and 39) are necessarily updated each time a read or write operation occurs. Within the scope of Appellants' claim

[1] *See* Pascal col. 8, ll. 32-51.

1, we broadly but reasonably construe a "memory" and a "cache" as encompassing any memory-based storage element, including a register. (Claim 1). We also note that Slane teaches the notoriously well known use of a cache. (FF 8).

Issues 2 and 3

Regarding Issues 2 and 3, we find the limitations of representative claims 2 and 4 are clearly met by the cited combination of Pascal and Slane because Slane expressly teaches that an "enqueue operation places new data at the tail 66 of the queue 62 and therefore does a memory write. The dequeue operation removes data from the head 64 of the queue 62, and therefore does a memory read." (FF 8). Therefore, we conclude it would have been immediately obvious to an artisan that only the head pointer (and not the tail pointer) would need be fetched and returned to memory when reading the queue in response to a dequeue operation (dequeue command). (Representative claims 2 and 4).

Issue 4

We decide the question of whether the Examiner erred by improperly combining the Pascal and Slane references under § 103. Appellants contend that the Examiner has failed to show a suggestion or motivation to combine the teachings of Pascal and Slane. (App. Br. 11-12).

At the outset, we observe that the Examiner merely looks to the secondary Slane reference as evidence that Appellants' claimed enqueue and dequeue operations were known to artisans at the time of Appellants' invention. (Ans. 4; *see also* FF 8). In this context, we view the relevant level of skill in the art as any person who has completed at least a first-year introductory computer science course pertaining to data structures.

Based on this record, we are of the view that Appellants' purported improvement over the prior art (regarding at least claims 1-30) represents no more than the predictable use of prior art elements[2] according to their established functions, and thus would have been obvious to one of ordinary skill in the art. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 417 (2007) ("[W]hen a patent 'simply arranges old elements with each performing the same function it had been known to perform' and yields no more than one would expect from such an arrangement, the combination is obvious.") (citing *Sakraida v. AG Pro, Inc.*, 425 U.S. 273, 282 (1976).

Moreover, given the breadth of Appellants' representative claims 1, 2 and 4, we are not persuaded that combining the respective familiar elements of the cited references in the manner proffered by the Examiner would have been "uniquely challenging or difficult for one of ordinary skill in the art" at the time of Appellants' invention (*see Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007) (citing *KSR*, 550 U.S. at 418)).

[2] *e.g.*, memory queues, head and tail pointers, caches, copying or moving pointers in memory, and the like.

Therefore, we find the Examiner's proffered combination of familiar prior art elements according to their established functions would have conveyed a reasonable expectation of success to a person of ordinary skill having common sense at the time of the invention.[3]

Lastly, we note that Appellants have not rebutted the Examiner's legal conclusion of obviousness by showing that the claimed combination of familiar elements produces any new function. Appellants have not provided any factual evidence of secondary considerations, such as unexpected or unpredictable results, commercial success, or long felt but unmet need.

Thus, when we take account of the inferences and creative steps that a person of ordinary skill in the art would have employed, we find the Examiner has articulated an adequate reasoning with a rational underpinning that reasonably supports the legal conclusion of obviousness. Therefore, we find Appellants' arguments unavailing regarding the combinability of the cited references for essentially the same reasons proffered by the Examiner in the Answer, and as further discussed above. Accordingly, we find the Examiner did not err by improperly combining the Pascal and Slane references under § 103.

---

[3] *See Medichem, S.A. v. Rolabo, S.L.*, 437 F.3d 1157, 1165 (Fed. Cir. 2006) (explaining that whether there is "a reasonable expectation of success in making the invention via" a combination of prior art elements is a question of fact).

Claims 1-30

For at least the aforementioned reasons (regarding Issues 1-4), we sustain the Examiner's obviousness rejection of claims 1-30 over the combination of Pascal and Slane.

Issue 5

Claims 31-37

We decide the question of whether the Examiner erred in finding under § 103 that the combination of Pascal and Slane would have taught or suggested fetching from the memory to a cache the count and one of either the head pointer or tail pointer within the meaning of representative claim 31.

We note that Pascal teaches the use of two count registers (31 and 33), where count register 31 indicates the number of available queue storage locations to be written to and count register 33 indicates the number of available queue storage locations to be accessed (read). (FF 7).

However, in contrast to the discussion above regarding Pascal's P and C indexes, here we find Pascal's count registers 31 and 33 (Fig. 3) are not copied or moved (i.e., suggestive of being fetched from memory to a cache). Instead, we find Pascal's count registers 31 and 33 are merely incremented. (*Id.*). Therefore, we find Pascal does not teach or suggest "fetching from the memory to a cache the count . . . ." (Claim 31). The Examiner has not established, nor do we find, that Slane overcomes the deficiencies of Pascal.

Because each of independent claims 31, 33, 34, and 36 require "fetching from the memory to a cache the count *and* one of either the head pointer or tail pointer" (in commensurate language), we reverse the Examiner's rejection of claims 31-37 (including associated dependent claims) for essentially the same reasons articulated by Appellants on page 15 of the Brief. (underlined added).

DECISION

We affirm the Examiner's § 103 rejection of claims 1-30.

We reverse the Examiner's § 103 rejection of claims 31-37.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a). *See* 37 C.F.R. § 1.136(a)(1)(2009).

ORDER

AFFIRMED-IN-PART

pgc

INTEL/BSTZ
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040